(12) UK Patent Application (19) GB (11) 2 404 126 (13) A

(43) Date of A Publication 19.01.2005

(21) Application No: 0423098.3

(22) Date of Filing: 17.01.2002

Date Lodged: 18.10.2004

(62) Divided from Application No 0201048.6 under Section 15(4) of the Patents Act 1977

(51) INT CL[7]: H04L 9/12 29/06

(52) UK CL (Edition X): H4P PDCSP PPEB U1S S2209

(56) Documents Cited: US 6215878 B1 US 6038322 A

(58) Field of Search: UK CL (Edition W) H4P INT CL[7] H04L Other: Online: WPI, EPODOC, JAPIO, INSPEC

(71) Applicant(s): Toshiba Research Europe Limited (Incorporated in the United Kingdom) 32 Queen Square, BRISTOL, BS1 4ND, United Kingdom

(72) Inventor(s): Timothy David Farnham Chan Yeob Yeun

(74) Agent and/or Address for Service: Marks & Clerk 57-60 Lincoln's Inn Fields, LONDON, WC2A 3LS, United Kingdom

(54) Abstract Title: **Secure communications using a secret key valid for a certain period and verified using a time stamp**

(57) This invention generally relates to secure communications links in which asymmetric cryptographic techniques are used to establish a secure link using symmetric cryptography. The secure link is established between sender and recipient by sending a message including a secret number, to be used for symmetric cryptography, with the message being digitally signed using a private key of the sender. The secret number is valid only for a certain time period, and the message also includes a time stamp which can be used by the recipient to verfiy that the number is still valid.

The secret number may be extracted by the recipient using the sender's public key, which may be obtained from e.g. a certificate. The secret number may be a Diffie-Hellman value.

Applications include communication between a mobile terminal and a server.

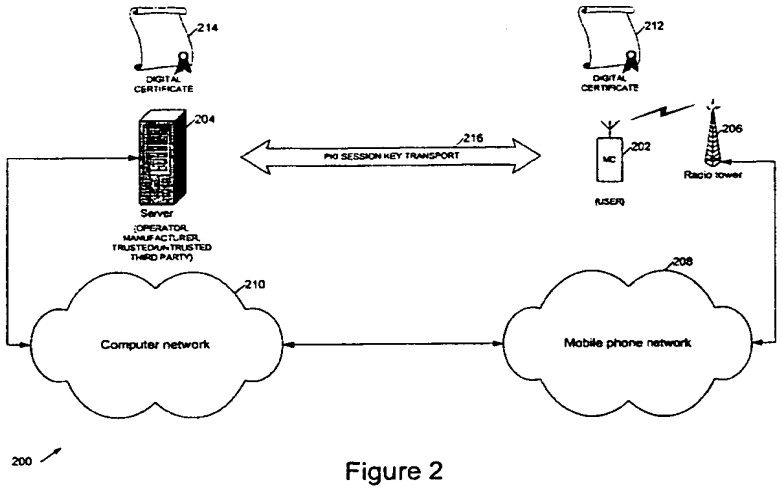

Figure 2

GB 2 404 126 A

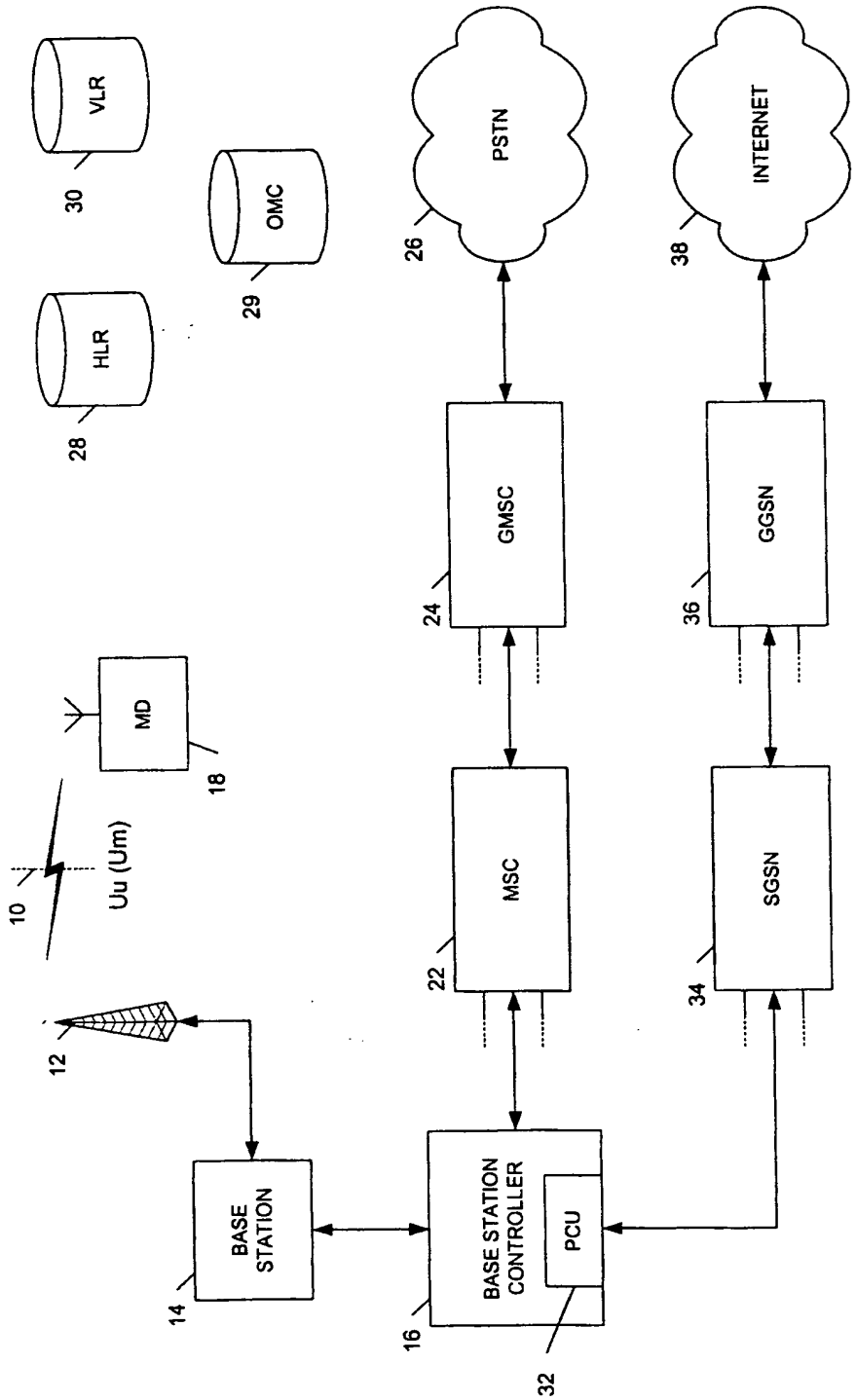

Figure 1
(PRIOR ART)

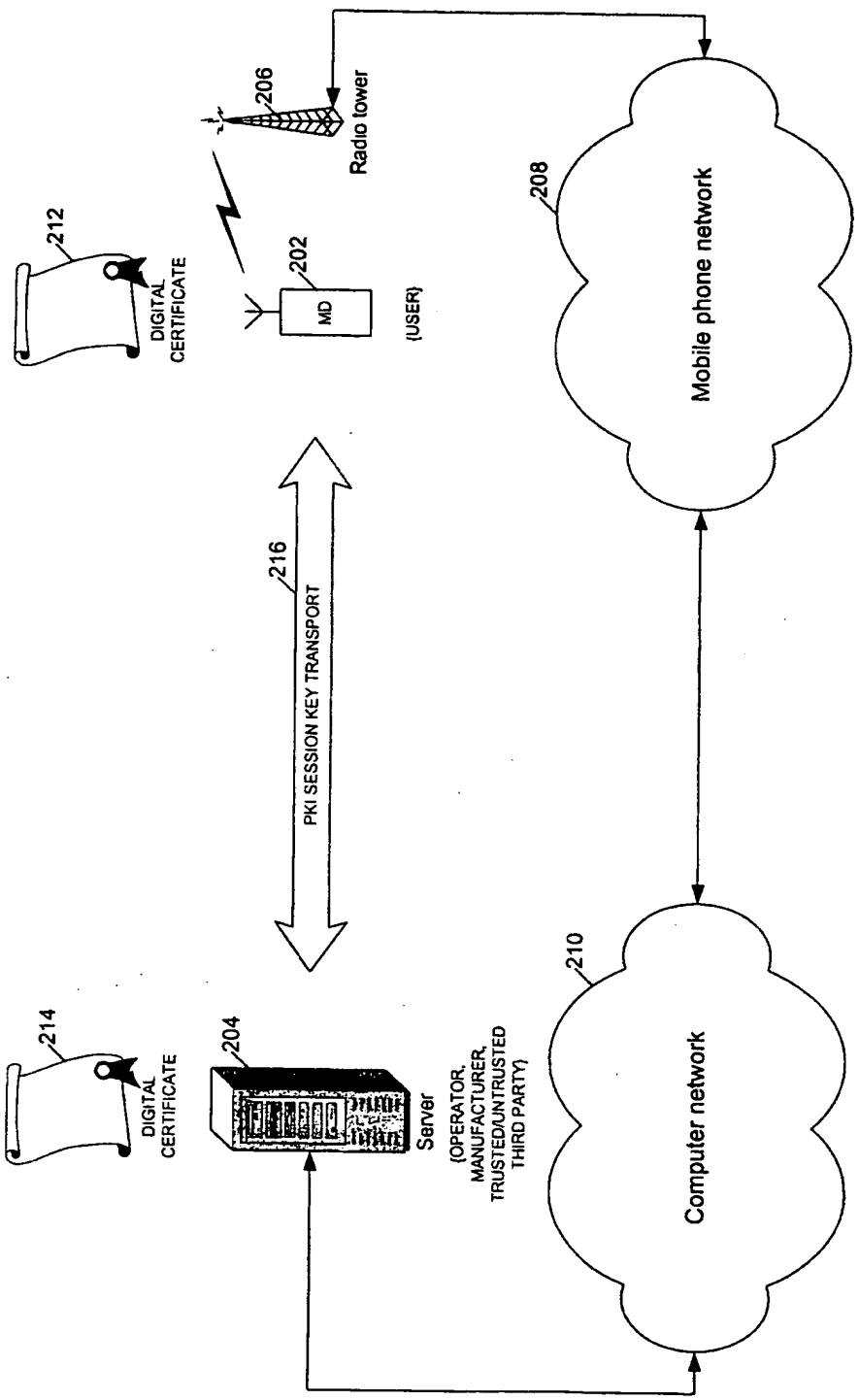
206
Radio tower
208
Mobile phone network
212
DIGITAL CERTIFICATE
202
MD
(USER)
216
PKI SESSION KEY TRANSPORT
214
DIGITAL CERTIFICATE
204
Server
(OPERATOR, MANUFACTURER, TRUSTED/UNTRUSTED THIRD PARTY)
210
Computer network
200

Figure 2

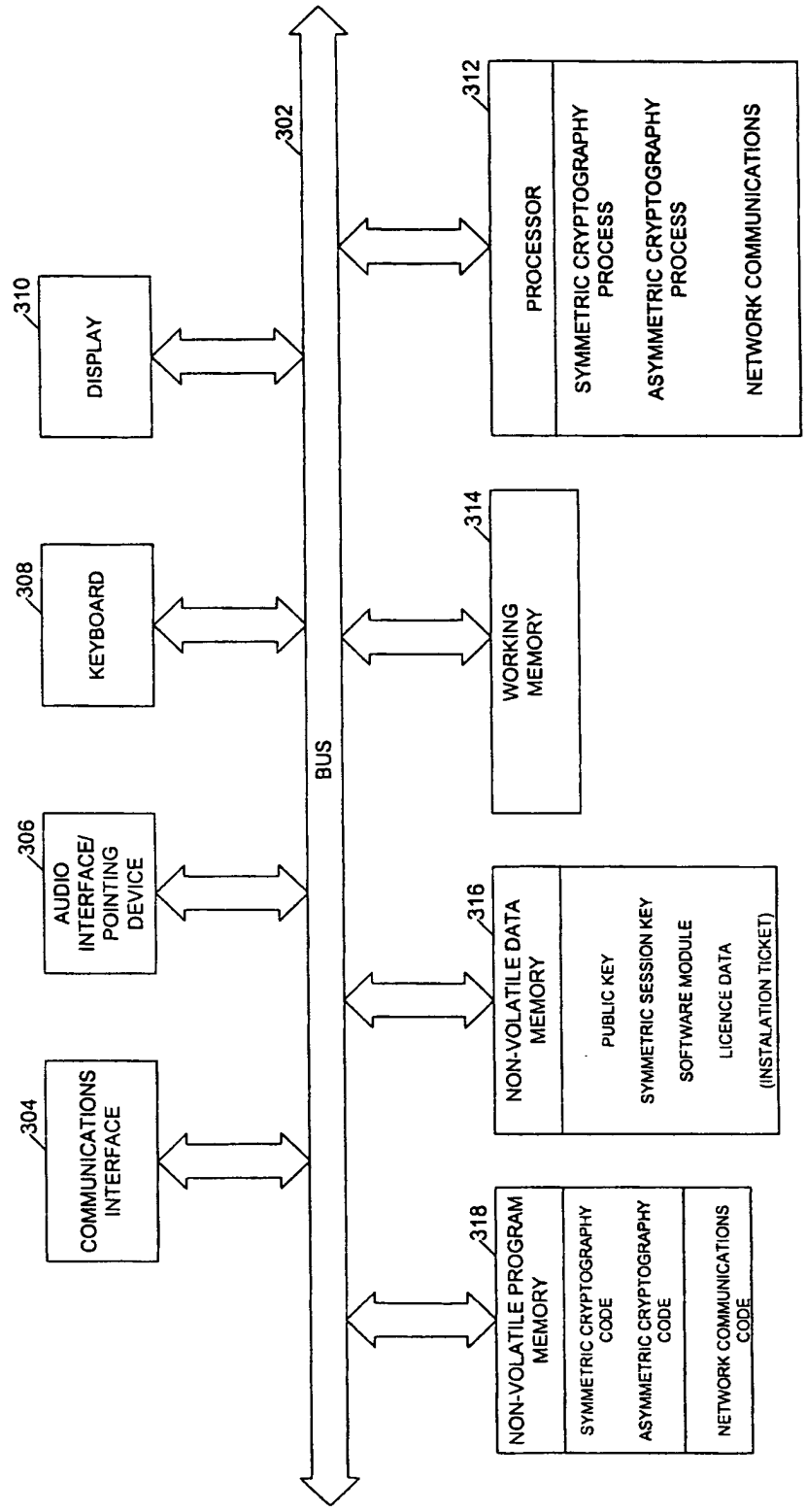
300
302
BUS
304
COMMUNICATIONS INTERFACE
306
AUDIO INTERFACE/ POINTING DEVICE
308
KEYBOARD
310
DISPLAY
312
PROCESSOR
SYMMETRIC CRYPTOGRAPHY PROCESS
ASYMMETRIC CRYPTOGRAPHY PROCESS
NETWORK COMMUNICATIONS
314
WORKING MEMORY
316
NON-VOLATILE DATA MEMORY
PUBLIC KEY
SYMMETRIC SESSION KEY
SOFTWARE MODULE
LICENCE DATA
(INSTALATION TICKET)
318
NON-VOLATILE PROGRAM MEMORY
SYMMETRIC CRYPTOGRAPHY CODE
ASYMMETRIC CRYPTOGRAPHY CODE
NETWORK COMMUNICATIONS CODE
320

Figure 3

DATA TRANSMISSION LINKS

This invention generally relates to secure communications links for data transmission and more particularly relates to data communications links in which asymmetric cryptographic techniques are used to establish a secure link using symmetric cryptography.

Data transmission is becoming increasingly important within mobile phone networks and, in particular, this is important to so-called 2.5G and 3G (Third Generation) networks as described, for example, in the standards produced by the Third Generation Partnership Project (3GPP, 3GPP2), technical specifications for which can be found at www.3gpp.org, and which are hereby incorporated by reference.

Secure data transmission is important for m-commerce but, in addition to this, the secure download and installation of software onto mobile terminals will also be important for multimedia entertainment, telle-medicine, upgrades for programmable mobile terminals, upgrades to different wireless standards, and the like. Reconfigurable mobile terminals are able to provide increased flexibility for end users who can customise the terminals for their personal needs by downloading and installing the desired applications, for example to support different types of radio systems and to allow the integration of different systems. However techniques are needed to protect mobile terminals against hackers maliciously substituting their software for software available from a handset manufacturer, network operator or trusted third party source.

Broadly speaking at present two basic cryptographic techniques, symmetric and asymmetric, are employed, to provide secure data transmission for example for software download. Symmetric cryptography uses a common secret key for both encryption and decryption, along traditional lines. The data is protected by restricting access to this secret key and by key management techniques, for example, using a different key for each transmission or for a small group of data transmissions. A well-known example of

symmetric cryptography is the US Data Encryption Standard (DES) algorithm (FIPS-46, FIPS-47-1, FIPS-74, FIPS-81 of the US National Bureau Standards). A variant of this is triple DES (3DES) in which three keys are used in succession to provide additional security. Other examples of symmetric cryptographic algorithms are RC4 from RSA Data Security, Inc and the International Data Encryption Algorithm (IDEA).

Asymmetric or so-called public key cryptography uses a pair of keys one "private" and one "public" (although in practice distribution of the public key is also often restricted). A message encrypted with the public key can only be decrypted with the private key, and vice-versa. An individual can thus encrypt data using the private key for decryption by any one with the corresponding public key and, similarly, anyone with the public key can securely send data to the individual by encrypting it with the public key safe in the knowledge that only the private key can be used to decrypt the data.

Asymmetric cryptographic systems are generally used within an infrastructure known as Public Key Infrastructure (PKI) which provides key management functions. Asymmetric cryptography can also be used to digitally sign messages by encrypting either the message or a message digest, using the private key. Providing the recipient has the original message they can compute the same digest and thus authenticate the signature by decrypting the message digest. A message digest is derived from the original message and is generally shorter than the original message making it difficult to compute the original message from the digest; a so-called hash function may be used to generate a message digest.

A Public Key Infrastructure normally includes provision for digital identity Certificates. To prevent an individual posing as somebody else an individual may prove his identity to a certification authority which then issues a certificate signed using the authority's private key and including the public key of the individual. The Certification Authority's public key is widely known and therefore trusted and since the certificate could only have been encrypted using the authority's private key, the public key of the individual is verified by the certificate. Within the context of a mobile phone network a user or the network operator can authenticate their identity by signing a message with their private key; likewise a public key can be used to verify an identity. Further details

of PKI for wireless applications can be found in WPKI, WAP-217-WPKI, version 24 - April 2001 available at www.wapforum.org and in the X.509 specifications (PKIX) which can be found at www.ietf.org, all hereby incorporated by reference.

In the context of 3G mobile phone systems standards for secure data transmission have yet to be determined and discussions are currently taking place in the MExE forum (Mobile Execution Environment Forum) at www.mexeforum.org. Reference may also be made to ISO/IEC 1170-3, "Information Technology – Security Techniques – Key Management – Part 3: Mechanism Using Asymmetric Techniques", DIS 1996.

Asymmetric cryptography was first publicly disclosed by Diffie and Hellman in 1976 (W. Diffie and D.E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22 (1976), 644-654) and a number of asymmetric cryptographic techniques are now in the public domain of which the best known is the RSA (Rivest, Shamir and Adleman) algorithm (R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, 21 (1978), 120-126). Other more recent algorithms including elliptic curve crypto systems (see, for example, X9.63, "Public key cryptography for the financial services industry: Key agreement and key transport using elliptic curve cryptography", Draft ANSI X9F1, October (1999)). The above-mentioned X.509 ITU (International Telecommunications Union) standard is commonly used for public key certificates. In this a certificate comprising a unique identifier for a key issuer, together with the public key (and normally information about the algorithm and certification authority) is included a directory, that is a public repository of certificates for use by individuals and organisations.

The main aims of a security system are authentication – of the data originator or recipient, access control, non-repudiation – proving the sending or reception of data, integrity of the transmitted data, and confidentiality. Preferably there should be provision for "anonymous" data download, that is the provision or broadcasting of data without specifically identifying a recipient.

The symmetric and asymmetric cryptographic techniques outlined above each have advantages and disadvantages. Asymmetric approaches are less resource-efficient, requiring complex calculations and relatively longer key lengths than symmetric approaches to achieve a corresponding level of security. A symmetric approach, however, requires storage of secret keys within the terminal and does not provide non-repudiation or anonymous software download. The present invention combines both these approaches, broadly speaking using public key techniques to transfer a secret session key. A symmetric session may then be established using this key, for example to download software securely. After software download this key may be stored in a repository in the mobile terminal for non-repudiation purposes or discarded once the software or other data download is complete. This technique supports a hierarchical infrastructure for key management such as X.509 or WPKI, the ability to broadcast to multiple mobile terminals, the ability to anonymously download software to mobile terminals (adopting asymmetric techniques) and faster software download by mobile terminals after establishing a symmetric session (using symmetric techniques).

According to one aspect of the invention there is therefore provided a method of establishing a secure communications link between a terminal and a server, the method comprising, assembling a message comprising a secret number and a digital signature for the secret number, the digital signature being generated using a private key for the server; encrypting the message at the server end of the communications link using a public key for the terminal; sending said encrypted message from the server to the terminal; decrypting said encrypted message at the terminal using a private key for the terminal; validating the message by checking the digital signature using a public key for the server; and establishing said secure communications link using said secret number; wherein the public and private keys for the terminal and server are public and private keys of an asymmetric cryptographic technique.

The secret number may either be sent alongside the digital signature or, where the signature is generated using an algorithm which allows message extraction, within the digital signature itself. The identity of the sender or recipient may be included within the message with, optionally, a time stamp or random number or nonce (as described above with reference to other aspects of the invention). Again the technique may be

employed where the establishment of the link is initiated by either the server or the terminal.

Thus, in another aspect, the invention provides a method of establishing a secure communications link between a server and a terminal, the method comprising: assembling a message comprising a secret number and a digital signature for the secret number, the digital signature being generated using a private key for the terminal; encrypting the message at the terminal end of the communications link using a public key for the server; sending said encrypted message from the terminal to the server; decrypting said encrypted message at the server using a private key for the server; validating the message by checking the digital signature using a public key for the terminal; and establishing said secure communications link using said secret number; wherein the public and private keys for the server and terminal are public and private keys of an asymmetric cryptographic technique.

A still further aspect of the invention relates to a method of establishing a secure communications link between a terminal and a server, the method comprising: performing, at the server-end of the communications link, a signing operation on a message comprising a secret number, using a private key for the server, to generate a digital signature, the message being recoverable from the digital signature; sending a message comprising the digital signature from the server to the terminal; extracting the secret number from the digital signature at the terminal and establishing said secure communications links using the secret number.

This technique complements that described above but allows the anonymous downloading of software and other data and is therefore usable, for example, for broadcasting a session key. Preferably an identification certificate for the server is stored in the terminal and the message includes an identifier for the server although this is not essential because, for example, the terminal may be pre-programmed to trust software from only one or a predefined group of sources.

In a yet further aspect the invention provides a method of establishing a secure communications link between a mobile terminal and a server, of a mobile

communications system, one of the terminal and server being an originator and the other a recipient, the method comprising: sending a first message from the originator to the recipient, the first message comprising: an identity certificate for the originator, the certificate including a public key for the originator, a first data block, and a signature of the originator generated by operating on the first data block, the first data block comprising at least an identifier for the originator and a secret number encrypted using a public key of the recipient; and authenticating the first message at the recipient using the originator identifier.

The originator identifier may be used, for example, to check the originator's signature. Again the technique may be employed where the establishment of the link is initiated by either the server or the terminal.

For convenience the method has been described as it applies to both ends of the communication link. However aspects of the invention provide separately only those steps of the method implemented at the server-end and only those steps implemented at the terminal end of the link.

In other aspects the invention provides computer program code to implement the method at the server-end of the link and computer program code to implement the method at the terminal-end of the link. This code is preferably stored on a carrier such as a hard or floppy disk, CD- or DVD-ROM or on a programmed memory such as a read-only memory or Flash memory, or it may be provided on an optical or electrical signal carrier. The skilled person will appreciate that the invention may be implemented either purely on software or by a combination of software (or firmware) and hardware, or purely in hardware. Likewise the steps of the method as implemented at either end of the link need not be necessarily be performed within a single processing element but could be distributed amongst a plurality of such elements, for example on a network of processors.

Embodimcnts of the above-described methods remove the necessity of installing a unique symmetric session key in the mobile terminal at manufacture and provide the ability to broadcast to multiple terminals and to provide anonymous software download

which is not otherwise achievable with symmetric techniques. The ability to anonymously download software and other data enables secure software and data download for each terminal/client request, thus enabling the downloading of free software, tickets, coupons and excerpts of a streamed media data such as music and MPEG movie clips. The combination of symmetric and asymmetric techniques, and in particular the ability of the methods to operate within an X.509 or WPKI infrastructure, also facilitates m-commerce. Furthemore the procedures are not entirely reliant on asymmetric techniques and allow, the faster symmetric algorithms also to be employed.

The skilled person will recognise that features and aspects of the above invention may be combined where greater security is required.

The invention will now be further described, by way of example only, with reference to the accompanying figures in which:

Figure 1 shows a generic structure for a 3G mobile phone system;

Figure 2 shows a schematic representation of key management for a secure communications link between a mobile device of a mobile phone network and a server coupled to the network; and

Figure 3 shows a computer system for implementing a method according to an embodiment of the present invention.

Figure 1 shows a generic structure of a third generation digital mobile phone system at 10. In Figure 1 a radio mast 12 is coupled to a base station 14 which in turn is controlled by a base station controller 16. A mobile communications device 18 is shown in two-way communication with base station 14 across a radio or air interface 20, known as a Um interface in GSM (Global Systems for Mobile Communications) networks and GPRS (General Packet Radio Service) networks and a Uu interface in CDMA2000 and W-CDMA networks. Typically at any one time a plurality of mobile devices 18 are attached to a given base station, which includes a plurality of radio transceivers to serve these devices.

Base station controller 16 is coupled, together with a plurality of other base station controllers (not shown) to a mobile switching centre (MSC) 22. A plurality of such MSCs are in turn coupled to a gateway MSC (GMSC) 24 which connects the mobile phone network to the public switched telephone network (PSTN) 26. A home location register (HLR) 28 and a visitor location register (VLR) 30 manage call routing and roaming and other systems (not shown) manage authentication, billing. An operation and maintenance centre (OMC) 29 collects the statistics from network infrastructure elements such as base stations and switches to provide network operators with a high level view of the network's performance. The OMC can be used, for example, to determine how much of the available capacity of the network or parts of the network is being used at different times of day.

The above described network infrastructure essentially manages circuit switched voice connections between a mobile communications device 18 and other mobile devices and/or PSTN 26. So-called 2.5G networks such as GPRS, and 3G networks, add packet data services to the circuit switched voice services. In broad terms a packet control unit (PCU) 32 is added to the base station controller 16 and this is connected to a packet data network such as Internet 38 by means of a hierarchical series of switches. In a GSM-based network these comprise a serving GPRS support node (SGSN) 34 and a gateway GPRS support node (GGSM) 36. It will be appreciated that both in the system of Figure 1 and in the system described later the functionalities of elements within the network may reside on a single physical node or on separate physical nodes of the system.

Communications between the mobile device 18 and the network infrastructure generally include both data and control signals. The data may comprise digitally encoded voice data or a data modem may be employed to transparently communicate data to and from the mobile device. In a GSM-type network text and other low-bandwidth data may also be sent using the GSM Short Message Service (SMS).

In a 2.5G or 3G network mobile device 18 may provide more than a simple voice connection to another phone. For example mobile device 18 may additionally or

alternatively provide access to video and/or multimedia data services, web browsing, e-mail and other data services. Logically mobile device 18 may be considered to comprise a mobile terminal (incorporating a subscriber identity module (SIM) card) with a serial connection to terminal equipment such as a data processor or personal computer. Generally once the mobile device has attached to the network it is "always on" and user data can be transferred transparently between the device and an external data network, for example by means of standard AT commands at the mobile terminal-terminal equipment interface. Where a conventional mobile phone is employed for mobile device 18 a terminal adapter, such as a GSM data card, may be needed.

Figure 2 schematically illustrates a model 200 of a system employing a method according to an embodiment of the present invention. A mobile device 202 is coupled to a mobile communications network 208 via a radio tower 206. The mobile communications network 208 is in turn coupled to a computer network 210, such as the Internet, to which is attached a server 204. One or both of the mobile device 202 and server 204 stores a digital certificate, the digital certificate 212 stored in mobile device 202 including a public key for server 204 and the digital certificate 214 stored in server 204 including a public key for the mobile device 202. (Other embodiments of the invention dispense with one or both these digital certificates).

A PKI session key transport mechanism 216 is provided to transport a session key between the mobile device 202 and the server 204, the PKI transport mechanism employing asymmetric cryptographic techniques using information from one or both of the digital certificates. The session key transported by the PKI mechanism is a secret session key for use with a symmetric cryptographic procedure and, because of the PKI transport, there is no need to store and manage pre-installed unique secret session keys on the server or mobile device.

The PKI transport mechanism 216 may comprise a unilateral transport mechanism from the server to the mobile device or vice-versa or may provide a mutual exchange mechanism for obtaining a shared session key. The server may be operated by a network operator, mobile device manufacturer, or a trusted or untrusted third party;

where the server is operated by an untrusted third party, the digital certificates may be dispensed with.

The mobile device is typically controlled by a user of the mobile communications network. For simplicity only a single mobile device is shown although, in general, a session key may be multicast to a plurality of such devices, or even broadcast.

Figure 3 shows a general purpose computer system 300 for implementing methods, as described below, according to embodiments of the invention. Depending upon whether the computer system is at the server end or the mobile user end of the link the computer system may comprise part of the server 204 of Figure 2 or part of the mobile device 202 of Figure 2. Where the computer system comprises part of the mobile device it may be implemented within the device itself or on a separate computer system attached to the device or in some other manner, for example on a SIM card or similar module.

The computer system comprises an address and databus 302 to which is coupled a keyboard 308, display 310 and an audio interface 306 in the case of a mobile phone or a pointing device 306 in the case of a server (unless the implementation is on a SIM card) in which case the phone provides these functions. Also coupled to bus 302 is a communications interface 304 such as a network interface (for a server), a radio interface (for a phone) or a contact pad interface (for a SIM card). Further coupled to bus 302 are a processor 312, working memory 314, non-volatile data memory 316, and non-volatile programme memory 318, the non-volatile memory typically comprising Flash memory.

The non-volatile programme memory 318 stores network communications code for the phone/server's SIM card operating system and symmetric and asymmetric cryptography code. Processor 312 implements this code to provide corresponding symmetric and asymmetric cryptography processes and a network communications process. The non-volatile data memory 316 stores a public key, preferably within a digital certificate, the server storing a public key for one or more mobile users, the mobile device storing public keys for one or more server operators. The non-volatile data memory also stores a symmetric session key, once this has been established, software (either for download

from the server or software which is being downloaded onto the mobile device/SIM card) and preferably licence data for the software and, in some instances, one or more installation tickets for controlling use of downloaded software. The software may comprise data such as video or MP3 data or code.

Generally it is desirable that software or data is obtained by a mobile terminal from trustworthy entities or trusted providers such as manufacturers, operators, and service providers that can be relied upon to make correct statements about the validity of software modules. The information that a trusted entity considers a specific core software module to be valid should preferably be made available to the terminal in a secure way.

In a symmetric approach a so-called ticket server issues installation tickets only for valid software modules. It is controlled and operated by a trusted provider. By issuing an installation ticket, the ticket-server represents that the software module which the ticket is referring to is valid. The installation ticket contains a cryptographically-strong, collision-resistant (hard to guess) one-way hash value of the software module which the terminal uses to check the integrity of the downloaded software module. A Message Authentication Code (MAC) (for example a keyed hash function see, for example, Computer data authentication. National Bureau of Standards FIPS Publication 113, 1985) is used to protect the installation ticket. This MAC is computed using a secret key shared by the terminal and the ticket server. By checking a ticket's MAC, the terminal verifies that a trusted provider has issued the ticket and that the ticket has not been modified. Then it checks the integrity of the received software module by comparing the hash values of the received software module and the one contained in the installation ticket. However, this technique does not guarantee non-repudiation in the event of any dispute between the trusted provider and the terminal users, since both shares the secret key so anyone who has the secret key could generate the MAC of a ticket.

An asymmetric signed license approach makes use of public-key cryptography. Similarly to the ticket-based approach, a license contains the information necessary to authenticate the integrity of a software module. A signed license can be a newly defined format, or it can be in previously defined format, such as an X.509 certificate, or a

WTLS (Wireless Transport Layer Security) certificate. A license should preferably at least contain the cryptographic hash of the software module and other pertinent information, such as validity dates, the issuer identity, and the recipient identity can also be included. The license is signed by a license server, which is controlled and operated by a trusted provider.

The license server issues licenses only for valid software modules, so by issuing a license for a piece of software, the license server in effect states that this software module is valid. Since a public-key signature scheme is used, every entity that has access to the public-key of the license server can check the signature of a license. Thus, this approach provides non-repudiation if there is any dispute between mobile terminal users and the service provider that will protect the both parties. In other words, only the license server can generate a valid signature for a license since only the license server knows the corresponding private key to sign the license.

Terminals can obtain an installation ticket or a signed license in different ways. They can wait until a software module is received and then directly ask for the ticket or license from the server. Alternatively, a ticket or license may be obtained indirectly through a download server or reconfiguration manager node. In the indirect approach, the software is bundled with the ticket or license and the entire package is sent to the terminal.

The symmetric and asymmetric approaches differ in the requirements they put on the terminal capabilities and on the amount of security data. The signed license approach requires that the terminal perform asymmetric cryptographic operations, which, in general, are more costly in terms of processing power and memory, which are in short supply on a terminal than symmetric cryptographic operations. The ticket-server approach requires only secret-key cryptography, which, in general, requires less processing. However, in the symmetric approach, communication with an online ticket server is always necessary, whereas with the asymmetric approach, it is not necessary for the license server to always be online.

In both cases, the terminal needs to compute the collision-resistant one-way hash value of the loaded software module. In the symmetric approach a ticket's validity is confirmed using a MAC, and in the asymmetric approach, a licence's validity is confirmed by checking a digital signature. A digital signature typically requires more data, so the number of bits in a license will generally be more than in a ticket.

The main objective of both these approaches is to protect terminals against malicious downloaded software. They do not protect against attacks that involve physical modifications of the terminal, such as the replacement of program memory, nor are they are intended to limit the distribution and use of software or to protect a software module against reverse-engineering. The security of the symmetric approach, however, requires that the terminal maintain the secrecy of the cryptographic key that it shares with the ticket server, whereas the asymmetric approach relies on a public-key, i.e. the level of secrecy required to protect the symmetric key is necessary for protecting the public key.

In this described embodiment to integrate the symmetric and asymmetric approaches it is assumed that PKI (Public Key Infrastructure) is employed and trusted parties such as manufacturers and operators issue their certificates to mobile terminals which store them in secure tamper resistance modules such as smart or other cards (for example, a SIM: Subscriber Identity Module, WIM: Wireless Identity Module, SWIM: Combined SIM and WIM, USIM: Universal Subscriber Identity Module).

PKI provides non-repudiation and protects both parties; the symmetric session key provides a low overhead and fast download once it has been transported (using the certified public key) from trusted parties such as manufacturers, operators, etc. This session key may be valid for only a short period for increased security.

This approach provides a unique secret session key so there is no need to install such a key, and no need for permanent secure storage of a key in the mobile terminal which otherwise can limit the key management between the trusted service providers and the terminals and the ability to broadcast to multiple mobile terminals and provide anonymous software download. The anonymous software download techniques for the

mobile terminal which will be described enable secure software download for each terminal/client request such as downloading free software, tickets, coupons and the like.

Firstly software download techniques initiated by the operator/server will be described. The originator *A* in this example the trusted software provider, (i.e. the terminal manufacturer, network operator, or the like is assumed to possess a priori an authentic copy of the encryption public key of the intended recipient *B*, the mobile terminal, and the terminal is assumed to have a copy of the server's (public) encrypting key.

One technique for establishing a shared secret session key is then as follows:

$$M1: A \rightarrow B: P_B(k||B||T_A||S_A(k||B||T_A||LC)) \qquad \text{Equation 1}$$

where $M1: A \rightarrow B$, denotes that *A* sends *M1* to *B*, and where *k* is a secret session key, *B* is an optional identifier for *B* (the intended recipient), T_A is an optional time stamp that is generated by *A*, LC is an optional digital licence, for example a software licence and || denotes concatenation of data. Utilising a time stamp hinders replay attacks, but in other embodiments a (preferably random) number may be used in addition to, or in place of, the time stamp, TH, for example generated from a clock. This may be used as a seed for a deterministic pseudo – random number generator so that both A & B can then generate synchronised series of pseudo-random numbers for use as session keys. Such a number (in the message) may be a nonce – a number used only once. $P_B(Y)$ denotes public key encryption such as RSA, (R.L. Rivest, A. Shamir and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, 21 (1978), 120-126). ECC, (N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, 48 (1987), 203-209) ElGamal, (T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, 31 (1985), 469-472) of data *Y* using party *B*'s public key and $S_A(Y)$ denotes a signature operation on *Y* using *A*'s private signature key.

Alternatively, a signature operation which allows recovery of the signed message can be used, such as the RSA signature with message recovery algorithm (ISO/IEC 9796, "Information technology – Security techniques – Digital signature scheme giving message recovery", International Organization for Standardization, Geneva, Switzerland, 1991) can be used as follows:

$$M1: A \rightarrow B: P_B(S_A(k||B||T_A||LC)) \qquad \text{Equation 2}$$

where k is a secret session key, B is an optional identifier for B (the intended recipient), T_A is an optional time stamp that is generated by A, and LC is an optional digital licence, for example a software licence.

In use, once the terminal obtains a signed session key, for example with a license, the terminal waits for a software module to arrive and, after receiving the software, the terminal is able (i.e. permitted) to execute the software with the session key. Alternatively, an entire software package can be sent to terminal together with a signed session key and license.

A related technique employing an anonymous RSA signature with message recovery can be used for downloading free software and coupons. This can be useful for trusted service providers wishing to broadcast trial versions of software and short clips of music and movies. In such cases it is desirable for anyone to be able intercept messages to obtain a session key. This key may be valid for only a short period for example 30 minutes for a film trailer reducing the need for authentication although it is desirable to provide for identification of the session key issuer, preferably an identification which can be easily verified. Thus the session key may be digitally signed by the manufacturer/operator or the service provider. One embodiment of this technique is therefore as follows:

$$M1: A \rightarrow B: S_A(k||B||T_A||LC) \qquad \text{Equation 3}$$

where k is a secret session key, B is an optional identifier for B (the intended recipient), T_A is an optional time stamp that is generated by A, and LC is an optional digital licence, for example a software licence.

In this embodiment an RSA signature operation with message recovery scheme is used (for example, ISO/IEC 9796:1991). Since the message is signed by A there is no need to include an identifier for A; including an identifier for the recipient allows the recipient to confirm they are the intended recipient. The terminals receiving *M1* each have an appropriate certificate for A, the originator/operator to allow the message to be extracted from S_A, for example, stored on SIM. This can also be used for broadcasting a session key to allow free software download, and enables terminals to download software anonymously.

In a variant of this technique, the key k is replaced by a Diffie-Hellman public value $g^n \bmod p$ (see, for example, W.Diffie and D.E. Hellman, *ibid*), where n is a positive integer satisfying $1 \le n \le p-2$. An alternative to *M1* is then as follows:

$$M1: A \rightarrow B: S_A(g^n \bmod p \| B \| T_A \| LC) \qquad \text{Equation 4}$$

where k is a secret session key, B is an optional identifier for B (the intended recipient), T_A is an optional time stamp that is generated by A, and LC is an optional digital licence, for example a software licence.

The mobile terminal B or the client can obtain the server's public value $Y_A = g^a \bmod p$ that is contained in the server key exchange or the SIM may contain the server's public value. The originator (in this example, the server A) chooses a random value n, computes $g^n \bmod p$ and sends *M1* including $g^n \bmod p$ to the terminal. The server A can then compute a session key $k = Y_A^n = (g^a)^n = g^{an} \bmod p$ and the terminal B can compute the same session key using $k = (g^n)^a = g^{na} \bmod p$.

Encrypted software may then be sent to the terminal *B* by encrypting the software with the common session key. An eavesdropper does not know the private key of server (that is *a*) and thus, it is computationally infeasible to determine the session key. This method can be used for distributing system software to mobile equipment for anonymous secure software download, for example for broadcasting a SIM update, because an individual recipient need not be specified.

In the above four scenarios, upon decrypting *M1*, recipient *B* will use a session key to download software from the originator/operator *A*. After software download, *B* may put the session key in the repository or may discard the session key which depends on the key management between the trusted service providers and the terminals.

In the above scenarios, upon decrypting *M1*, the recipient *B* can use the session key to download software from the originator/operator *A*. After the software download, *B* may put the session key in the repository or may discard the key, which is chosen depending on, among other things, the key management between the trusted service providers and the terminals. For an operating system upgrade a non-anonymous, rather than an anonymous technique is preferred as it is useful to know to whom the upgrade has been sent.

Next software download techniques initiated by the mobile terminal will be described; these are close to mirror images of the above server-initiated techniques. We will describe a secure software download and anonymous software download techniques based on asymmetric techniques such as RSA and Diffie-Hellman, for initiating key changes from the mobile terminal. These techniques can be used for establishing a symmetric session key for secure implementation of each individual request for a data item or group of items, such as software, tickets, coupons, and the like.

In the technique signed bocks are encrypted by combining a digital signature and public key encryption as follows:

$$M1: B \rightarrow A: \; P_A(k \| A \| T_B \| S_B(k \| A \| T_B \| LC)) \qquad \text{Equation 5}$$

where k is a secret session key, A is an optional identifier for A (the intended recipient), T_B is an optional time stamp generated by B, and LC is an optional digital licence, for example a software licence.

The terminal, *B*, generates a session key and signs a combination of the session key, *A*'s identity and a time stamp. This session key, signature and, optionally the time stamp and A's identifier, are encrypted with the server's certified public key extracted, for example, from a prior server key exchange message. Software, such as video clips and music, is sent from the server *A* to the client *B* using the session key. Since an eavesdropper does not know the server's private key, it is computationally infeasible for him/her to compromise the session key *k*, particularly since this may be only valid for one session or a limited period.

As previously described an anonymous crytographic technique such as anonymous RSA can also be described, as follows:

$$M1: B \rightarrow A: P_A(k||A||T_B||LC) \qquad \text{Equation 6}$$

where k is a secret session key, A is an optional identifier for A (the intended recipient), T_B is an optional time stamp generated by B, and LC is an optional digital licence, for example a software licence.

The terminal, *B* generates a session key *K* and encrypts it with the server's certified public key (extracted from a server key exchange message). The software may then be sent to the client *B* using the session key *K*. Since an eavesdropper does not know the server's private key, it is computationally infeasible for the one time session key *k* to be compromised.

Alternatively, an anonymous Diffie-Hellman cryptographic technique can be employed as follows (a mobile-initiated technique is described; the server-initiated technique corresponds):

First an appropriate prime p and generator g of Z_p^* are selected and published, and, for example, stored on the terminal SIM. Here Z_p^* is the multiplicative group 1,2,3.... p-1 and $(2 \leq g \leq p-2)$. One way to generate an appropriate p and g is described in RFC (Request For Comments) 2631.

$M1: B \rightarrow A: \ g^b \bmod p$ Equation 7

The mobile terminal B or client can obtain the server's public value $Y_A = g^a \bmod p$ where is the private key of the server, for example from a server key exchange. Preferably, however the server's public value is stored in the SIM. The terminal chooses a random value b, computes $g^b \bmod p$ and sends $M1$ $g^b \bmod p$ (encrypted) to the server. Both a and b are positive integers satisfying $1 \leq a \leq p-2$ and $1 \leq b \leq p-2$. The mobile terminal B can compute a key for a symmetric session $k = Y_A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$ and the server A can compute the same session key $k = (g^b \bmod p)^a \bmod p = g^{ba} \bmod p$. Encrypted data or software may then be sent to the terminal B by encrypting it with a session key or the session key may be used by both the terminal and server to generate another common key, for example by operating on data known to both with K. An eavesdropper does not know the private key of server (a) and it is thus computationally infeasible to determine the session key. Anonymous RSA and Diffie-Hellman can be used, for example for downloading free software, tickets and coupons.

Anonymous software download techniques generally only provide protection against passive eavesdroppers. An active eavesdropper or active man-in-the-middle attack may replace the finished message with their own during the handshaking process for creating sessions. In order to avoid this attack server authentication is desired.

Analogously to the anonymous RSA signature technique with message recovery described above with reference to Equation 4, the Diffie-Hellman value $g^b \bmod p$ may be encrypted using the originator's (that is, in this example, B's) private key. More specifically it may be protected by sending the Diffie-Hellman value as a digital

signature from which the signed message is recoverable. The recipient may then recover $g^b \bmod p$ using the originator's public key, more specifically by extracting the message from the signature.

Under certain circumstances, the Diffie-Hellman and (DH) the related Elliptic Curve Diffie-Hellman (ECDH) key agreement schemes (X9.63, "Public key cryptography for the financial services industry: Key agreement and key transport using elliptic curve cryptography", Draft ANSI X9F1, October (1999)) are susceptible to a class of attacks known as "small-subgroup" attacks. Where, if a key belongs to a small subgroup a directed brute-force attack based on guessing keys from the subgroup may succeed. In the anonymous DH and ECDH cases there is a risk that such a small subgroup attack will lead communicating parties to share a session key which is known to an attacker. This threat can be alleviated by using a predetermined group determined "good" or "strong" values of g and p and checking that received public keys do not lie in a small subgroup of the group, or by not re-using ordinary DH key pairs. Background information on protection against these attack, can be found in the draft ANSI standards X.9.42 (X.9.42, "Agreement of symmetric keys using Diffie-Hellman and MQV algorithms", ANSI draft, May (1999)) and. X.9.63 (X9.63, "Public key cryptography for the financial services industry: Key agreement and key transport using elliptic curve cryptography", Draft ANSI X9F1, October (1999)).

Mutual key authentication protocols will now be described. In these both *A* and *B* are authenticated by exchanging messages having information or a property characteristic of *A* and *B*, in the protocols below messages encrypted using the public keys of *A* and *B*.

In a first mutual authentication process *A, B* possess each other's authentic public key or, each party has a certificate carrying its own public key, and one additional message is sent by each party for certificate transport to the other party. Background information on this protocol can be found in Needham and Schroeder (R.M. Needham and M.D. Schroeder, "Using encryption for authentication in large networks of computers", Communications of the ACM, 21 (1978), 993-999).

The messages sent are as follows:

$M1: A \rightarrow B: \ P_B(k_1 || A || T_A)$ Equation 8

$M2: A \leftarrow B: \ P_A(k_1 || k_2)$ Equation 9

$M3: A \rightarrow B: \ P_B(k_2)$ Equation 10

The steps of the procedure are as follows:

1. *The originator operator (or server) A* sends *M1*, including a first key k_1, to *B*.
2. *The receiver user (terminal) B* recovers k_1 upon receiving *M1*, and returns *M2*, including a second key k_2, to *A*.
3. Upon decrypting *M2, A* checks that the key k_1 recovered from *M2* agrees with that sent in *M1*. *A* then sends *B M3*.
4. Upon decrypting *M3, B* checks the key k_2 recovered from *M3* agrees with that sent in *M2*. The session key may be computed as $f(k_1 || k_2)$ using an appropriate publicly known non-reversible function *f* such as MD5 (Message Digest 5, as defined in RFC 1321) and SHA-1 (secure Hash Algorithm-1, see, for example, US National Bureau of Standards Federal Information Processing Standards (FIPS) Publication 180-1.
5. *B* then starts downloading software by using the symmetric session key $f(k_1 || k_2)$. After software download, *B* may discard the session key or keep it for a short period, depending on the key management strategy.

A second X509 mutual authentication process operates in the context of the X.509 strong two-way authentication procedure (ISO/IEC 9594-8, "Information technology – Open systems interconnection – The directory: Authentication framework", International Organisation for Standardization, Geneva, Switzerland 1995) is described as follows:

Let $D_A = (T_A || \ R_A || B || \ P_B(k_1))$, $D_B = (T_B || \ R_B || A || \ P_A(k_2))$. Equation 11

Where *A* and *B* comprise identifiers for the server and terminal respectively.

$M1: A \rightarrow B: Cert_A \| D_A \| S_A(D_A)$ Equation 12

$M2: A \leftarrow B: Cert_B \| D_B \| S_B(D_B)$ Equation 13

Where the $Cert_A$ and $Cert_B$ are public certificates for A & B respectively. The steps of the procedure are as follows:

1. A obtains a timestamp T_A indicating an expiry time, then generates a random number R_A, obtains a symmetric key k_1, encrypts K_1, using P_B and sends a message $M1$ to B. (Since the message is signed by A there is no need to include an identifier for A; including an identifier for the recipient in D_A allows the recipient to confirm they are the intended recipient).
2. B verifies the authenticity of $Cert_A$, extracts A's signature public key, and verifies A's signature on the data block D_A. B then checks that the identifier in $M1$ specifies itself as intended recipient and that the timestamp T_A is valid, and checks that R_A has not been replayed.
3. If all checks succeed, B declares the authentication of A successful, decrypts k_1 using its a session key, and saves this now shared key for downloading software securely. (This terminates the protocol if only unilateral authentication is desired.). B then obtains a timestamp T_B, generates random number R_B, and sends A a message $M2$.
4. Similarly A carries out actions analogous to those carried out by B. If all checks succeed, A declares the authentication of B successful, and key k_2 is available for subsequent use. A and B share mutual secrets k_1 and k_2 so the session key may be computed as $f(k_1 \| k_2)$ which may then be used for downloading software securely (here "software" is used in a general sense to mean soft data).

An authenticated Diffie-Hellman session key exchange can be achieved by using public key encryption as follows:

The originator A (that is the trusted software provider, terminal manufacturer, operator or the like) and a mobile terminal B possess an authentic copy of the encryption public

key of A and B this may be, for example, locally stored or the public keys may be exchanged between the parties, for example, as digital certificates. As with anonymous Diffie-Hellman described above an appropriate prime p and generator g of Z_p^* $(2 \le g \le p-2)$ are selected and published and, preferably, stored locally in the terminal messages are then exchanged as follows:

$$M1: A \rightarrow B: P_B(g^a \bmod p \| A \| T_A) \qquad \text{Equation 14}$$

$$M2: A \leftarrow B: P_A(g^b \bmod p \| B \| T_A \| T_B) \qquad \text{Equation 15}$$

$$M3: A \rightarrow B: S_A(E_k(software \| LC)) \qquad \text{Equation 16}$$

Where A & P_A and B and P_B comprise identifiers and public keys of the originator and terminal respectively and T_A and T_B are time stamps for messages from A & B respectively (A, B, T_A and T_B are optional) k denotes an encryption operation performed using key k.

A chooses a random value a, computes $g^a \bmod p$ and sends $M1$ to B (there is no need to store g^a mod p in the terminal and because this value is encrypted it is safe from main-in-the-middle attacks). The mobile terminal B decrypts the received message using its private key and chooses a random value b, computes $g^b \bmod p$ and sends $M2$ ($g^b \bmod p$) to A which decrypts the message using its private key. Both a and b are positive integers satisfying $1 \le a \le p-2$ and $1 \le b \le p-2$. The terminal B then computes a session key $k = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$; the originator A can also compute the session key using $k = (g^b \bmod p)^a \bmod p = g^{ba} \bmod p$. A then signs the encrypted software and LC preferably using the shared session key k and sends it to B; here LC is a software licence, optionally specifying a validity period of the session key k, giving copyright details and the like. An eavesdropper does not know the private keys of A and B and commitment values a and b. and thus, it is computationally infeasible to determine the session key and the threat from man in the middle attacks is alleviated. The encrypted identifiers A and B provide a guarantee of the sender's identity for the messages, thus preferably $M1$ includes A although there is less need for $M2$ to include B. Similarly only B knows T_A so including this in $M2$ (whether or not T_B is also

included) allows A to imply that the message was correctly received by B. Including T_B permits a time window $T_B - T_A$ to be defined; this is preferably shorter than any likely decrypt time, for example less than one hour. Here, preferably T_A defines a sending time for *M1* and T_B a receive time (at B) for *M1*.

In variants of the method alternatives to *M3* are as follows:

i) $M3 : A \rightarrow B \;\; E_k$ *(software ||LC)*

ii) $M3 : A \rightarrow B \;\; E_k$ *(software ||LC)* S_A *(*E_k *(software ||LC))*

iii) $M3 : A \rightarrow B \;\; E_k$ *(software)* S_A *(LC)*

These alternatives can provide faster encryption. In (ii) a signature operation without operation message recovery can be used; in (iii) only the licence is signed, preferably with message recovery, unless the licence is within the software (optionally in (iii) an encrypted version of the licence E_k(LC) may be signed).

Timestamps may be used to provide freshness and (message) and can provide a time window for uniqueness guarantees, message reply. This helps provide security against known-key attacks is required, vulnerable to replay attacks of the unilateral key authentication protocols. The security of timestamp-based techniques relies on use of a common time reference. This in turn requires that synchronised host clocks be available and clock drift and must be acceptable given the acceptable time window used. In practice synchronisation to better than 1 minute is preferred although synchronisation to better than 1 hour may be acceptable with longer time windows. Synchronisation can be achieved by, for example, setting an internal clock for the terminal on manufacture.

Where the terminal possesses an authentic certificate for A, the originator or operator, (either locally stored or received in a message) then the above unilateral key authentication techniques provide secure software download. For mutual authentication protocols where both A and B possess authentic certificates or public keys there are no known attacks which will succeed, apart from brute force attacks to recover the private keys of A and B. However in an X.509 – context proccdure, because there is no

inclusion of an identifier such as A within the scope of the encryption P_B within D_A, one cannot guarantee that the signing party actually knows the plaintext key. That is, because the identity is not encrypted the message could be signed by someone who had not encrypted the key.

The uses of public key technology to transport a symmetric session key for secure software download has been described. This combines the advantages of both the asymmetric and symmetric approaches. PKI provides with non-repudiation and protects both parties if there is a dispute, but PKI is computationally intensive and would be inefficient for secure software download on its own. A symmetric session key provides a means to enable efficient and fast download once the key has been transported using a certified public key issued by trusted parties. The lifetime of the session key can be short (for example for a single data transfer) or long (for example, months) depending on the security requirements and likelihood of the key being compromised.

The described techniques are also suitable for the MExE standard for future programmable mobile user equipment. Moreover, the anonymous software download techniques enable secure software download for each terminal/client request for downloading free software, tickets, coupons, as well as for secure M-Commerce.

Embodiments of the invention have been described in the context of a server and mobile terminal of a mobile communications system but aspect of the invention also have other applications, for example in networked computer systems. It will also be recognised, in general, either the terminal or the server may comprise the initial message originator in the above protocols although for conciseness the specific exemplary embodiments are described with reference to one or other of these as the orginator. The invention is not limited to the described embodiments but encompasses modifications apparent to those skilled in the art within the spirit and scope of the claims.

CLAIMS:

1. A method of establishing a secure communications link between a terminal and a server, the method comprising:

performing, at the server-end of the communications link, a signing operation on a message comprising a secret number, using a private key for the server, to generate a digital signature, the message being recoverable from the digital signature;

sending a message comprising the digital signature from the server to the terminal;

extracting the secret number from the digital signature at the terminal; and

establishing said secure communications links using the secret number, wherein the secret number is valid for a time period and wherein the message further comprises a time stamp, the method further comprising checking the validity of said secret number using the time stamp and establishing said secure communications link dependent upon the result of said checking.

2. A method as claimed in claim 1 wherein the message further comprises an identifier for the server, the method further comprising:

retrieving from storage in the terminal an identification certificate for the server including at least a public key for the server; and

using the server public key to extract said secret number.

3. A method of establishing a secure communications link between a server and a terminal, the method comprising:

performing, at the terminal-end of the communications link, a signing operation on a message comprising a secret number using a private key for the terminal to generate a digital signature, the message being recoverable from the digital signature;

sending a message comprising the digital signature from the terminal to the server;

extracting the secret number from the digital signature at the server; and

establishing said secure communications links using the secret number, wherein

the secret number is valid for a time period and wherein the message further comprises a time stamp, the method further comprising checking the validity of said secret number using the time stamp and establishing said secure communications link dependent upon the result of said checking.

4. A method as claimed in claim 1, 2 or 3 wherein the secret number comprises a Diffie-Hellman value $g^n \bmod p$, where p is a prime number and g is a generator for a Diffie-Hellman key exchange protocol and n is a positive integer less than p-1.

5. A data transmission link configured to implement the method of any one of claims 1 to 4.

6. A carrier carrying computer program code for a terminal to implement the part of the method of any one of claims 1 to 4 performed at the terminal end of the communications link.

7. A terminal including the carrier of claim 6.

8. A carrier carrying computer program code for a server to implement the part of the method of any one of claims 1 to 4 performed at the server end of the communications link.

9. A server including the carrier of claim 8.

28

Application No: GB0423098.3 **Examiner:** Mr Steven Davies

Claims searched: 1-9 **Date of search:** 9 November 2004

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
A	-	US 6215878 B1 (HARKINS)
A	-	US 6038322 A (HARKINS)

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC[W] :

H4P

Worldwide search of patent documents classified in the following areas of the IPC[07]

H04L

The following online and other databases have been used in the preparation of this search report

Online: WPI, EPODOC, JAPIO, INSPEC